HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 10990633-1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Lawrence Bain et al.

Confirmation No.:

Application No.: 09/298,~~953~~ 453

Examiner: Cong Lac T. Huynh

Filing Date: 04/13/1999

Group Art Unit: 2178

Title: DYNAMIC-ADAPTIVE CLIENT-SIDE IMAGE MAP

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Sir:

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on **May 17, 2005**.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) $500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

( ) (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d) for the total number of months checked below:

( )	one month	$120.00
( )	two months	$450.00
( )	three months	$1020.00
( )	four months	$1590.00

( ) The extension fee has already been filled in this application.

(X) (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account **08-2025** the sum of **$500.00**. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

(X) I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450. Date of Deposit: 7-18-05

OR

( ) I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number_____ on _____

Number of pages:

Typed Name: Joanne Bourguignon

Signature: Joanne Bourguignon

Respectfully submitted,

Lawrence Bain et al.

By Robert W. Bergstrom

Robert W. Bergstrom

Attorney/Agent for Applicant(s)
Reg. No. **39,906**

Date: July 18, 2005

Telephone No.: **206.621.1933**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent application of:

Inventors: Lawrence Bain et al.
Serial No. 09/298,953
Filed: April 13, 1999
For: DYNAMIC-ADAPTIVE CLIENT-SIDE IMAGE MAP

Examiner: Cong Lac T. Huynh
Group Art Unit: 2178
Docket No. 10990633-1
Date: January 16, 2004

BRIEF ON APPEAL

Commissioner for Patents
Alexandria, VA 22313-1450

Sir:

This appeal is from the decision of the Examiner, in a Final Office Action mailed on March 18, 2005, finally rejecting claims 1-24.

REAL PARTY IN INTEREST

Hewlett-Packard Development Company, L.P. is the Assignee of the present patent application. Hewlett-Packard Development Company, L.P., is a Texas corporation with headquarters in Houston, Texas.

RELATED APPEALS AND INTERFERENCES

Applicants' representative has not identified, and does not know of, any other appeals of interferences which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

STATUS OF CLAIMS

Claims 1-24 are pending in the application. Claims 1-24 were finally rejected in the Office Action dated March 18, 2005. Applicants' appeal the final rejection of claims 1-24, which are copied in the attached Appendix I.

STATUS OF AMENDMENTS

An Amendment After Final is enclosed with this brief. Amendments have been made to the claims to correct a number of typographical errors and to place the claims into proper form for the Appeal.

SUMMARY OF CLAIMED SUBJECT MATTER

Overview

The invention is described, below, with reference to Figures 1-5 of U.S. Application No. 09/298,953 ("Current Application"). Figure 1 shows a display of an example web page, a description of which is transmitted through the Internet from a remote server computer to a user's client computer. The example web page display includes a "Read This First" text line 106, a "Quotes, News, Charts, Data" text line 108, and a number of additional charts and figures 102 and 110-113. Commonly, the information used by the user's client computer to display a web page is encoded within an HTML file transferred through the Internet to the client computer by a remote server computer. The HTML file may include references to image files and other information that is separately accessed by the client computer from the server computer and integrated with the display. For example, the displayed-chart image 110 may be encoded within a separate file, with indications of the identity of the separate file and the position at which the image encoded in the separate file is to be displayed within the web-page display encoded within the HTML file.

At the time that the Current Application was filed, as discussed in the Background of the Invention section and in the first few pages of the Detailed Description of the Invention section of the Current Application, several relatively recent innovations in web-page display were coming into general use, including active regions and dynamic images. Active regions are regions within the display of a web page that are associated with actions, such as displaying a different web page or image, or invoking a software routine. The active regions are specified by one of two types of image maps: (1) a client-side image map that is

generated and maintained on the user's client computer; and (2) a server-side image map that is maintained on a remote server. An image map can be thought of as the superposition of an abstract template over an image included on the web page, with the abstract template containing descriptions of the size, shape, and location of each active region as well as an association between the active region and an action. Figure 2 shows a larger scale representation of the left-hand, vertical column portion of the displayed web page shown in Figure 1, and Figure 3 shows a representation of an image map for the left-hand, vertical column portion of the displayed web page that includes specification of two active regions 302 and 304 corresponding to the "Read This First" text line (216 in Figure 2) and the "Quotes, News, Charts, Data" text line (218 in Figure 2). When a user employs a mouse or other input device to position a cursor displayed on the display device within an active region, the display of the cursor may change to indicate that further input of a mouse click, or other type of input, will invoke display of a different web page or image. For example, a mouse click input after positioning of a cursor over the "Read This First" text line (216 in Figure 2), specified by the image map active region 302, may result in display of a text file with additional user information overlying the initially displayed web page.

At the time that the Current Application was filed, locations within a displayed image were generally designated by device-relative coordinates. For a computer display, imaginary horizontal and vertical axes incremented by pixels or by arbitrary horizontal and vertical unit increments are used to assign two-dimensional coordinates to each point in the display. In Figure 2, for example, the corners of the left-hand, vertical column portion of the displayed web page are labeled with device-relative coordinates, such as device-relative coordinates (0,0) that label the top, left-hand corner, and in Figure 3, the corners of the template and of the active regions 302 and 304 are also labeled with device coordinates. Device-relative coordinates are also used by the operating system of a client computer to specify the location of a cursor moved by mouse input. Therefore, the position of a cursor can be correlated with a displayed image by comparing the device-relative coordinates specifying the current cursor position with device-relative coordinates specifying, among other things, active regions within the displayed image. For example, a routine running either on the client computer or on the remote server can determine that a mouse click is meant to activate display of a text file associated with active region 302 in Figure 3 by comparing the device-relative coordinates corresponding to the current cursor position with the device-relative coordinates specifying the corners of active region 302. Figure 4 shows the image map of Figure 3

superimposed over the left-hand, vertical column portion of the displayed web page shown in Figure 2, as an illustration of the correspondence between the image map and the displayed web page which is obtained from the common use of device coordinates in both the image map and the displayed web page.

Server-side image maps involve a browser, running on a client computer displaying a web page, transmitting, using device coordinates, user input events that arise during display of the web page back to the server computer, followed by transmission of responses from the server computer to the client computer that facilitate any actions invoked by the user input. Client-side image maps, by contrast, are implemented, at run time, by web browsers, or by viewers invoked by web browsers, on client computers. Input to displayed images associated with client-side image maps and execution of corresponding actions thus does not incur the extra Internet traffic incurred in displaying images associated with server-side image maps.

Dynamically sizable and shiftable images within web pages are another example of relative recently developed enhanced graphic capabilities for web pages at the time that the Current Application was filed. A user viewing the display of a dynamic image within a web page may direct the browser to zoom into or out from the image to higher and lower resolutions, and to scroll the image in vertical and horizontal directions. The software mechanisms for describing and implementing dynamic images within web pages were initially separate and distinct from the mechanisms for describing and implementing client-side image maps.

At the time that the Current Application was filed, active-region functionality was not well integrated with dynamic image functionality. As a result, when client-side image maps were associated with dynamic images within web pages, the active regions were not automatically zoomed and shifted, or panned, along with the dynamic images. As a result, the active regions defined within the client-side image maps quickly lost their original correspondence to specific regions of the images with which they were associated. For example, as shown in Figure 5, when the scale of display of the left-hand, vertical column portion of the displayed web page 502 is enlarged, via a zoom operation, the image map 504 is not automatically correspondingly enlarged, leaving the specified active regions no longer overlying the portions of the displayed web page that they are intended to overlie. As a result, a mouse click input when the displayed cursor overlies the word "This" in the "Read This First" text line 510 would not result in the desired display of a text file, as it would prior to the zoom operation, as shown in Figure 4.

As discussed in the Summary of the Invention section of the Current Application, embodiments of the present invention address the problem discussed in the previous paragraph by providing methods and systems for associating active regions with positions within displayed images in a device-and-display independent manner. In the Current Application, an embodiment of the present invention is described that represents an implementation of a dynamic-adaptive client-side image map that automatically tracks changes in a displayed dynamic image within a web page in order to maintain the originally specified correspondence between active regions defined by the dynamic-adaptive client-side image map and regions of the associated image displayed as part of the web page. In addition, the dynamic-adaptive client-side image map definitions are accessible to software routines invoked from web pages. In this embodiment of the present invention, an adaptive delivery module running on a server computer determines the capabilities of a client computer requesting a web page and, when the requesting computer's capabilities are compatible with inclusion of a dynamic-adaptive client-side image map, the adaptive delivery module transforms the HTML description of the requested web page to include one or more dynamic-adaptive client-side image maps and transmits the transformed HTML description of the web page to the requesting client computer. An enhanced web browser running on the requesting client computer receives the transformed web page and instantiates an appropriate image viewer based on tags within the transformed HTML description of the web page. The web browser provides the instantiated viewer with parameters describing any dynamic-adaptive client-side image maps within the transformed HTML description of the web page, and the instantiated viewer then processes the parameters to produce data structures that describe active regions within the image displayed by the viewer as part of display of the web page. The information contained within the data structures can be used at run time, during the display of the web page, by the instantiated viewer to correlate user input with active regions of the displayed image.

The Current Application provides a detailed, pseudo-code description of an embodiment of the client-side image viewer, encapsulated, in part, in the pseudo-code class "Image." As described in the Current Application beginning on line 34 of page 25:

> An Image object contains data members that represent the device coordinates of the upper left corner and lower left corner of the displayed images, "leftX," "leftY," "rightX," and "rightY" declared above on lines 4-7. In the current implementation, displayed images are assumed to be rectangular. These values are represented as

> floating point values to allow for precise multiplication by scale factors for zooming operations. An Image object also contains a reference to the image, "img" declared above on line 8, and a reference to a MapInfo object, "map" declared above on line 9 that represents a DACSIM associated with the image. An instantiated Image object, in the current implementation, is an instantiated enhanced viewer.

One aspect of the described embodiment is the use, by the client-side image viewer invoked by a user's web browser on the user's client computer, of image-relative coordinates, as described beginning on line 5 of page 20 of the Current Application:

> The x,y image coordinates are stored as floating point values in data members "x" and "y" declared above on lines 4 and 5. Image coordinates range from 0 to 1. They are fractions of the width and height of an image, and are thus valid regardless of how the image is scaled or translated in device coordinates. These fraction coordinates are one embodiment of device-independent, image-relative coordinates, referred to as image coordinates.

As described in the Current Application, beginning on line 14 of page 19, the client-side image viewer employs instantiated objects of the class "Point" to represent defining points of active regions, and the class "Point," in turn, employs image-relative coordinates.

Thus, as described in the above-quoted passages from the Current Application, the described embodiment of a client-side image viewer stores image dimensions in device-relative coordinates, and stores the locations and positions of active regions in image-relative coordinates that are fractions of the image widths and heights. The image-relative coordinates therefore provide an automatic correspondence between active regions and displayed images. The client-side image viewer relates device coordinates to active regions using image-relative coordinates and by transforming the position of a mouse click, received from the operating system of the client computer in device coordinates, into image-relative coordinates prior to searching the active regions for an active region overlying the position of the mouse click, as described in the Current Application beginning on line 16 of page 25:

> The Image function "mouseClick" takes the device coordinate position of a mouse click input event, "x" and "y," as arguments and invokes an appropriate action if the mouse click was input to any active region within the image. First, on line 6, mouseClick determines whether the mouse click was input within the image. If so, then mouseClick sets the values of the local variables "imageX" and "imageY" to the image-relative coordinates of the mouse click, on lines 9 and 10, above. Then, in the *while* loop of lines 11-19, mouseClick determines whether the mouse click event was directed to any of the active regions via a call to the ImageArea member function "isInside" through an AreaInfo pointer stored in loop variable "a," on line

13, and, if so, invokes the mouse click action associated with the active region on line 15.

Independent Claim 1.

Aspects of Applicants' invention discussed in preceding paragraphs are clearly and straightforwardly claimed in claim 1. Applicants claim a method for associating an active region with a position within an image included in a page displayed by a client-side web browser. The client-side browser requests a page for display from a server, and receives, in response to the request, a description of the page that includes an invocation of a client-side image viewer for displaying the image and that includes parameters that are passed to the image-viewer, upon invocation, that describe the image to be displayed and a client-side image map used to correlate active regions within the image to the displayed image. Upon instantiation of the client-side image viewer on the client computer, the client-side viewer stores representations of active images within the image in image-relative coordinates, as discussed in the above overview section. The client-side image viewer employs those representations to correlate input events passed from the browser to the image viewer with active regions, so that an input event directed to an active region within an image is properly handled.

Dependent Claims 2 – 10

Claim 2 further specifies the displayed page as a web page. Claim 3 specifies Internet communications linking a client and server. Claim 4 specifies an inter-process communications medium linking a server and a browser. Claim 5 specifies than an HTML document is displayed. Claim 6 specifies that an OpenPix image is displayed. Claim 7 specifies types of user input and actions to be taken in response to user input. Claim 8 further specifies the nature of image-relative coordinates. Claim 9 species detection of an event by a browser and altering the display by the viewer. Claim 10 species events that include zoom and pan.

Independent Claim 11

Aspects of Applicants' invention discussed in preceding paragraphs are clearly and straightforwardly claimed in claim 11. Applicants claim a method for serving a description of a page from a server to a browser running on a client computer that requests the page, the

description of the page provided to the browser by the server containing an invocation of a viewer, the invocation including parameters that specify an image included in the page and an active region within the image. The server receives a request from the browser for a description of the page that includes a specification of the image and an associated client-side image map, the client-side image map specifying a shape, size, and location of the active region within the image and that specifies actions to be performed in response to input events directed to the active region. In response to the received request, the server retrieves a description of the page and determines the capabilities for viewing pages provided by the browser running on the client computer. When the browser is capable of accepting display altering commands from a user while displaying a page, the server parses the description of the page to find the specification of the image and the client-side image map included in the page, substitutes, in the description of the page, an invocation of a viewer for the specification of the image and the client-side image map included in the page, including in the invocation parameters that specify the image and the client-side image map, to create a transformed page description. Finally, the server sends the transformed page description to the browser

Dependent Claims 12 – 17

Claim 12 further specifies the displayed page as a web page. Claim 13 specifies Internet communications linking a client and server. Claim 14 specifies an inter-process communications medium linking a server and a browser. Claim 15 specifies than an HTML document is displayed. Claim 16 specifies that an OpenPix image is displayed. Claim 17 specifies types of user input and actions to be taken in response to user input.

Independent Claim 18

Aspects of Applicants' invention discussed in preceding paragraphs are clearly and straightforwardly claimed in claim 18. Applicants claim system for displaying a page that includes an image and an active region correlated with a particular portion of the image, the display of the page modifiable during the display of the page on a display device of a client computer such that the active region within the image remains correlated with the portion of the image. The system comprises a browser running on the client computer that displays the page, a server that receives a request from the browser for a description of the page and that that provides a description of the page that contains an invocation of a viewer, the invocation

including parameters that specify an image included in the page and an active region within the image, and data structures on the client computer that store image-relative indications of the particular portion of the image associated with the active region and actions and actions to be performed in response to input events directed to the active region.

Dependent Claims 19 – 24

Claim 19 further specifies the displayed page as a web page. Claim 20 specifies Internet communications linking a client and server. Claim 21 specifies an inter-process communications medium linking a server and a browser. Claim 22 specifies than an HTML document is displayed. Claim 23 specifies types of user input and actions to be taken in response to user input. Claim 24 species additional details about image-relative coordinates.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether claims 1-24 are unpatentable under 35 U.S.C. 103(a) over Guedalia, U.S. Patent No. 6,356,283 B1 ("Guedalia") in view of Beri et al., U.S. Patent No. 6,141,018 ("Beri").

ARGUMENT

Claims 1-24 are currently pending in the application. In an Office Action dated July 15, 2003 ("Office Action"), the Examiner maintained a prior rejection of claims 1-24 under 35 U.S.C. § 103(a) as being unpatentable over Guedalia, U.S. Patent No. 6,356,283 B1 ("Guedalia"), in view of Beri et al., U.S. Patent No. 6,141,018 ("Beri"). Applicants respectfully traverse these rejections.

Prior to presenting detailed arguments with respect to the above identified issues, the cited references Guedalia and Beri first summarized, below:

Guedalia

Guedalia is discussed, in this and in following paragraphs, with reference to Figures 1-6 of Guedalia. In the Abstract, Guedalia summarizes his disclosure:

> A method and system for archiving digital data on a server computer, and enabling a user, by means of a client computer, to interactively view a digital image derived from the

request dispatched from the client to the server, which the server can parse to extract the selected command. (Guedalia, column 4, lines 36-41)

A key feature of the present invention is that it operates without the use of special client software, other than an Internet browser which is already resident on the client computer. (Guedalia, column 4, lines 52-55)

In Figures 3 and 4, Guedalia shows flow charts that describe a preferred embodiment of his invention. Beginning on line 56 of column 19, Guedalia describes the client computer sending mouse-pointer coordinates to a server for interpretation of the mouse-pointer coordinates with respect to the displayed page containing subregions:

> Using a mouse or keyboard 34 for positioning, the user clicks on the mouse at a position within the view window. HTML identifies the image portion being viewed as an image map, and the client sends the mouse pointer coordinates 52 to the server computer 10. The server computer 10 calculates the sub-region within which the mouse coordinates are situated, and dynamically creates a new HTML page 54, with a link to an embedded image, constituting the response to the user mouse click.

Similarly, on line 31 of column 20, Guedalia describes the client computer sending mouse-pointer coordinates to a server for interpretation of the mouse-pointer coordinates with respect to the displayed page containing subregions:

> At step 72 the user viewing the page clicks on a location within the view window that contains the embedded image portion. The browser extracts the mouse pointer coordinates where the user clicked at step 74, and sends them to the server at step 76.
>
> At step 78 the server receives the mouse pointer coordinates, and at step 80 the server calculates which sub-region contains these coordinates.

An alternative embodiment is described with reference to Figures 5 and 6. In this alternative embodiment, the user's browser interprets mouse clicks, rather than the server computer, as described beginning on line 24 of column 21:

> At step 112 a user clicks on one of several buttons, each corresponding to a specific interactive command. Examples of such commands are (1) centered zoom-in, (2) centered zoom-out, (3) pan in one of several directions, (4) reset to initial image view and (5) print. At step 114 the browser identifies the area where the user clicked, thereby identifying the button and the desired associated user command. At step 116 the client sends the command data to the server in the form of an HTTP request.

However, Guedalia does not discuss or disclose how the user's browser interprets the mouse pointer coordinates or how the user identifies which command button of the displayed page a mouse click is input to. There is also no indication that the displayed command buttons can

digital data, the method including receiving an original HTML page by the client computer from the server computer, the original HTML page containing a view window within which a first image is displayed, the view window being partitioned into a plurality of sub-regions at least one of which contains a multiplicity of pixels, selecting a location within the view window corresponding to one of the plurality of sub-regions by the user, initial sending by the client computer to the server computer an indication of the sub-region selected by the user, creating by the server computer a new HTML page containing a link to an embedded image which corresponds to the indication, and further sending the new HTML page by the server computer to the client computer.

The above-described method and system is clearly reflected in the Figures. For example, Guedalia discloses a method and system involving server-side rendering and support for dynamic HTML pages, clearly shown in Figure 1 by the location of the software for dynamic HTML generation 16 located on the server computer 10. Figure 2 illustrates the active regions of an image display according to Guedalia. A viewing window 42 is partitioned into five sub-regions, labeled in Figure as "Sub-region 1," "Sub-region 2," etc. When a user inputs a mouse click into a Sub-region of the viewing window, the user's browser, which displays the viewing window, sends mouse coordinates corresponding to the input click to a server. The server interprets the mouse-coordinates as corresponding to a sub-region, and then returns a magnified image corresponding to the sub-region for display by the user's browser, providing the effect to the user of interactively zooming into a selected portion of the originally displayed image.

The following quotes from Guedalia summarize important aspects of Guedalia's invention, with portions of the quotes underlined and/or italicized for emphasis:

> The term "client-less" refers to the absence of special client software. (Guedalia, column 3, lines 38-39)
>
> The present invention provides a novel approach to clientless HTML-driven interactive navigation over the Internet. (Guedalia, column 4, lines 2-4)
>
> Whenever the user clicks on the image being displayed, the browser sends the mouse pointer coordinates back to the server. (Guedalia, column 4, lines 13-15)
>
> The server's role is simply to identify which of these images is to be displayed in response to a specific interactive user command, and to prepare that image in a form that the browser recognizes, such as JPEG. (Guedalia, column 4, lines 24-27)
>
> The present invention also enables a client to interact with an image without the use of image maps, by selecting one of several commands, such as "zoom-in," "zoom-out," "pan," and "reset." The command selected is embedded within an HTTP

be resized or moved. When the client side-browser interprets the mouse click and sends the command correlated with that interpretation, as, for example, for zooming out a particular image, in steps 50 and 110 in Figure 5, the server computer sends an entirely new HTML page, in step 54 of Figure 5, without including the magnified image, and the client computer must then explicitly request the magnified image from the server, in step 58 of Figure 5, which the server returns in step 60 of Figure 5 for display by the client-side browser in step 62 of Figure 5. In Figure 6, these steps are numerically labeled 114, 116, 84, 96, 108, 94, respectively. *Therefore, Guedalia's client-side browser does not need to maintain a correspondence between an image map specifying active regions and displayed images. Guedalia's client-side browser receives an entirely new web page and new images displayed along with the web page after each command or operation invoked by a user's mouse input. Only the server needs to maintain a correspondence between active regions and the image, and that, in fact, is the entire point of Guedalia's disclosure.*

Guedalia does mention relative coordinates beginning on line 14 of column 24. Guedalia mentions that portions of requested FlashPix images may be requested by a client-side browser from a server using relative coordinates. For example, Guedalia states, beginning on line 66 of column 23:

> Specifically, this IIP request is for data from a FlashPix image file photo.fpx using version 1.0 of the IIP protocol. The desired image data corresponds to a 300x300 pixel resolution image of the entire region of the FlashPix image, in the JPEG format. ... When the user clicks on this image, an HTTP request is dispatched to IP address 123.456.789.01 and port 8081. Port 8081 is the server port that handles incoming data. ... The server knows how to interpret this string as follows: ... The third data item, 0.0,0.0,1.0,1.0,300,300,0.0,0.0,1.0,1.0 is the current region of the FlashPix image being displayed. Regions are specified as relative coordinates x_1, y_1, x_2, y_2 where (x_1, y_1) are the coordinates of the lower left corner of the region ...

However, Guedalia does not teach, mention, or suggest storing image maps specifying active regions in terms of image-relative coordinates or transforming mouse pointer coordinates into image-relative coordinates. The client-side browser does not need to do so. Instead, the client-side browser can use device-relative coordinates for all operations, since the client-side browser in Guedalia need not maintain a correspondence between active regions and displayed images. The browser in Guedalia need not maintain such a correspondence, because Guedalia receives a new HTML page description and new images for display following each operation or command solicited by mouse input. Moreover, the relative coordinates are not coordinates of an active region relative to a displayed image, but are,

instead, simply a scaled coordinate system for the image itself. In other words, instead of specifying a portion of an image in terms of pixels, the relative coordinates allow for specifying a portion of an image in terms of fractions of the horizontal and vertical edges of a rectangular image. But the image relative coordinates of Guedalia to do not specify the location of a separate entity, such as a separate graphic, contained within, or superimposed on, the displayed image. Moreover, Guedalia does not discuss resizing or moving the image over-all image or display, but only zooming and navigation functions carried out on portions of an image displayed within an image display. The only active regions are portions of the image, and those are defined by and recognized by the server, rather than the client. There are no active regions that need to be kept in correspondence with the image by the client computer, or the browser running on the client computer.

Finally, Guedalia's technique does not depend on specialized functionality embedded in the client-side browser. Instead, the server computer bears the burden of performing all image modifications needed for client-side, user-invoked, image-altering commands, such as zoom-in, zoom-out, and pan, by sending a new HTML page and new images to the client computer following each such operation initiated on the client compute by user input.

In summary, Guedalia does not disclose or suggest dynamic-adaptive client-side image maps as claimed in the current application. Guedalia instead discloses a clientless image navigation system that allows a user interacting with a client computer to selectively zoom into portions of a displayed image. However, each zoom operation is carried out on a remote server, and an entirely new HTML page is returned by the server upon each zoom operation. There is no teaching or suggestion in Guedalia of resizing the entire display, or changing the area of the display on a monitor, and no need for a client computer to maintain any correspondence between active regions within the displayed image and the image, because, again, each zoom or pan operation returns an entirely new HTML document. By contrast, a currently claimed dynamic-adaptive client-side image map allows a browser and image viewer on a local computer to modify the display of an image locally, by resizing the image or relocating the image, without invoking a server, and to maintain a correspondence between active regions within the displayed image and the image despite local image-modifying operations.

Beri

Beri discloses a method for displaying an HTML document in an animated marquee

(Beri, Abstract, lines 1-2). A marquee window is invoked in response to detection, by an HTML viewer, of a <marquee> tag within the HTML document. The <marquee> tag is generally accompanied by attributes, or parameters, that specify characteristics of the marquee window, including the display size, the URL of an HTML document that the marquee window is to display, the direction and speed of scrolling, and other parameters. The marquee window then displays the specified HTML object in a scrolling fashion that provides a visual effect that is similar to the electronic marquees found on buildings that display the headline news (Beri, column 2, lines 1-9 and column 5, lines 29-31). The sequence of steps leading to display of an HTML document in a marquee window are shown in Figure 5 of Beri. A client-side web browser 501 invokes an HTML viewer 502 to display an HTML document 504. When the HTML document 504 contains a <marquee> tag, the HTML viewer instantiates a marquee object 503. The HTML viewer 502 then sets attributes of the marquee object, including the URL of an HTML document for the marquee object to display. The marquee object 503 then invokes a second instantiation of an HTML viewer (both viewers are for some reason labeled 503 in Figure 5) to generate the image (Beri, column 5, lines 20-38).

The parameters furnished by the HTML viewer to control marquee display by the marquee object include a "Zoom" parameter and a "WidthOfPage" parameter. As disclosed in Beri on lines 58-65 of Beri:

> The Zoom parameter indicates a percentage by which the image being scrolled is to be reduced/enlarged. The one-to-one aspect ration [sic] is maintained on "zoomed" pages. A value of -1 indicates the image is resized to fit in the marquee window (one-to-one aspect ration [sic] not necessarily kept as a result).
>
> The WidthOfPage parameter specifies the width in pixels to format image.

The marquee object notifies the HTML viewer when various events occur. These events include the start of image display, the end of image display, and indication of the bounds, indication of a scroll, or an indication that the left mouse button has been pressed (Beri, column 6, lines 66 through column 7, line 3). An "OnLMouseButtonDown" event is issued when the left mouse button is pressed in the marquee window (Beri, column 7, lines 33-35). This would allow the HTML viewer to treat the marquee window, as a whole, as an active region, but not subregions of the window, or portions of the image displayed within the marquee window. In other words, mouse clicks input to the marquee window are detectable, and an HTML viewer displaying a marquee via a marquee object can receive the input mouse clicks, pass them to a browser, and the browser can respond in some fashion to them by

passing the events to an HTML-viewer handler. The marquee object provides an interface to the HTML viewer that renders an HTML for display in the marquee window, allowing the HTML viewer to determine various parameters of the marquee window, such as the width of the display area, in pixels (Beri, column 8, lines 2-12 and 45).

Beri does not once mention client-side image maps or image-relative coordinates. None of the events generated by the marquee object relate to dynamic resizing, and there is nothing indicated in Beri that would necessitate image-relative coordinates in order to maintain a correspondence between active regions of an image with the display of the image. It is almost axiomatic that the image displayed in the marquee window does not contain active regions. It would be difficult for a user to accurately position a mouse cursor over an active region of a rapidly scrolling image. Images displayed in the marquee window are generally only informational or artistic, and there is no suggestion in Beri for active regions within the displayed image. All sizing and display parameters are specified to the marquee object by the invoking HTML viewer, and all widths and coordinates are specified in pixels, or, in other words, in device coordinates. Presumably, were the marquee object's size or position to be changed, the invoking HTML viewer would either destroy the current marquee object and instantiate a new marquee object with new parameters, or adjust the parameters of the currently running marquee object. However, again, all such parameters specify sizes and coordinates in device-relative coordinates, or pixels, and there is no mention or suggestion in Beri of resizing operations. Also, Beri's marquee object does not handle events, but instead passes events back to a browser for handling.

Arguments Directed to Issue (1)

1. Whether claims 1-24 are unpatentable under 35 U.S.C. 103(a) over Guedalia, U.S. Patent No. 6,356,283 B1 ("Guedalia") in view of Beri et al., U.S. Patent No. 6,141,018.

Claim 1 of the Current Application is representative of the current claims, and is first discussed in detail with respect to the cited references and rejection under 35 U.S.C. 103(a). The remaining claims that either depend from claim 1, or that include language similar to the language of claim 1, are discussed, below, more concisely.

Applicants' position with regard to the cited references can be summarized with reference to claim1:

(1) First, claim 1 is directed to "[a] method for associating an active region with a corresponding position within an image included in a page displayed by a browser running on a client computer. Guedalia and, arguably, Beri disclose active regions within a larger image displayed by a browser, and, in both, an active region is kept in static correspondence with a displayed image. However, the invention of claim 1 is directed to keeping an active region in dynamic correspondence with a displayed image, as revealed in the elements of the claim, discussed below.

(2) Second, claim 1 is directed to "sending a request by the browser to a server for a description of a page that includes a specification of the image and an associated client-side image map, the client-side image map specifying a shape, size, and location of the active region within the image and specifying actions to be performed in response to input events directed to the active region." Guedalia, as discussed above, does not teach, mention, or suggest client-side image maps that specify a shape, size, and location of active regions within an image displayed on the client computer and actions to be performed in response to input events directed to the active regions. However, it can be reasonably assumed that Guedalia may, in fact, employ something equivalent to a static client-side image map, because the client does recognize mouse click input to portions of the displayed image that are resizable within the over-all displayed image. On the other hand, Guedalia explicitly states that the server interprets the mouse clicks based only on the supplied, device-relative mouse-cursor coordinates transmitted to the server by the browser on a client computer. Guedalia does obliquely refer to an alternative embodiment in which a client-side browser interprets mouse clicks input to command buttons, but provides no teaching or suggestion on how this is accomplished. Guedalia does not mention or suggest that the over-all display is resizeable or even moveable. Because Guedalia is primarily directed to server-side processing, it is conceivable that the client-side browser employs server support for interpreting mouse clicks input even to these command buttons. Alternatively, the browser may store the command button regions as device coordinates, along with the device-coordinated-based specification of everything else which the browser displays. Guedalia's client-side browser does not decide on actions, or execute actions, corresponding to user input, but simply sends mouse coordinates of the input to the server for interpretation and execution. The server generates a new HTML page description in response to receiving the mouse coordinates. Again, Guedalia emphasizes that the image-navigation system in clientless, and does not involve specialized, client-resident executables or functionality added

to a client-side browser. Thus, in Appellant's representative's respectfully offered opinion, the first element of claim 1 does not read on Guedalia's disclosure.

Beri does not teach, mention, or suggest transferring client-side image maps to a viewer or browser. An invoking viewer can specify the width of the page in which a marquee object is to be displayed, using the *WidthOfPage* parameter, but the marquee object apparently decides, based on that one parameter, how to display itself. Moreover, the *WidthOfPage* parameter is specified in device-relative coordinates and furnished as parameter when invoking a marquee-object or setting parameters through a marquee-object interface, and not supplied in a client-side image map. Beri does not teach or suggest resizing of the marquee window or the over-all image in which it is displayed, and therefore does not suggest any reason for maintaining a correspondence between active regions within a displayed image and the over-all image. The marquee object does not translate mouse clicks input to the marquee display window, but instead forwards them back to the invoking browser for interpretation and execution of corresponding actions, and, therefore, no actions are specified for active regions to the marquee object in a client-side image map or by any other means. Thus, the first element of claim 1 is neither taught, mentioned, nor suggested in Beri.

(3) Third, claim 1 is directed to "receiving from the server in response to the request a description of the requested page that includes an invocation of a viewer for displaying the image, the invocation including parameters that describe the image and the client-side image map." Guedalia does not discuss invocation of a viewer, or parameters that describe a client-side image map. Beri does not mention or suggest client-side image maps, and does not mention or suggest a parameter that specifies to an invoked marquee object where to obtain or access a client-side image map. Thus, the second element of claim 1 is neither taught, mentioned, nor suggested in either Guedalia or Beri.

(4) Fourth, claim 1 is directed to "instantiating the viewer and passing to the viewer the parameters included in the invocation." The reference "the parameters" refers to "parameters that describe the image and the client-side image map" of the previous element. While Beri does teach invoking of a viewer, Beri does not teach passing parameters that describe a client-side image map. Guedalia does not mention or suggest instantiation of a viewer, and does not therefore specify a client-side image map to a viewer. Thus, the third element of claim 1 is neither taught, mentioned, nor suggested in either Guedalia or Beri.

(5) Fifth, claim 1 is directed to "storing by the viewer representations of active regions within the image in image-relative coordinates along with indications of the actions to be performed in response to input events directed to the active region." Neither Guedalia nor Beri teaches, mentions, or suggests storing, by a client-side browser, or any client-side entity, representations of active regions in image-relative coordinates. Thus, the fourth element of claim 1 is neither taught, mentioned, nor suggested in either Guedalia or Beri.

(6) Sixth, claim 1 is directed to "when an input event is detected by the browser during display of the page, passing the input event by the browser to the viewer." Guedalia does not teach, mention, or suggest a client-side viewer to which events may be passed. Beri does teach detection of input events, but the input events are detected not by the client-side browser, but instead by the marquee-object invoked by an HTML viewer, in turn invoked by the browser. Input events are passed by the marquee object back to the HTML viewer instantiated by the client-side browser. Neither Guedalia nor Beri teach, mention, or suggest detection of input by a client-side browser and passing of these events to a viewer. Thus, the fifth element of claim 1 is neither taught, mentioned, nor suggested in either Guedalia or Beri.

(7) Finally, claim 1 is directed to "when the viewer determines that the input event was input to a position within the image corresponding to the active region, determining an action specified for performance in response to the input event to the active region and calling for performance of the determined action." Guedalia does not mention or suggest a client-side viewer. Beri teaches that the marquee object determines when a mouse click is input to the marquee window, and passes the event back to the HTML viewer, and that, when the HTML viewer includes an appropriate event handler for the event, the client-side browser invokes the event handler. Thus, the sixth element of claim 1 is neither taught, mentioned, nor suggested in either Guedalia or Beri.

In summary, neither Guedalia nor Beri teach, mention, or suggest a single one of the second through sixth elements of claim 1. Guedalia and Beri both involve static association of active regions within displayed images to the over-all image, while the currently claimed invention is directed to dynamic association of active regions within displayed images to the over-all image, as revealed in the elements of claim 1. There is no basis for a 35 U.S.C. § 103(a) rejection of claim 1 based on either or both of Guedalia and Beri.

Response to Arguments in Office Action

In the Office Action, the Examiner states that Guedalia discloses "sending a request by the browser to a server for a description of a page that includes a specification of the image and a size, and location of active region within the image and specifying actions to be performed in response to input events directed to the active region." Guedalia does not disclose this. As discussed above, Guedalia is a clientless image navigation system. All actions to be performed in response to input on the client are made by the server. The client sends mouse coordinates of an input click to the server, in step 76 of Figure 4, and the server returns an entire new HTML page for display by the client. Any differences between the previously displayed HTML page and the new HTML page are effected entirely by the server in response to the received mouse coordinates.

In the Office Action, the Examiner states that Guedalia discloses "receiving from the server in response to the request a description of the requested page." Indeed, the server sends an HTML page to the client for display by the client browser.

In the Office Action, the Examiner states that Guedalia discloses "passing the input event by the browser to the viewer when the input event is detected by the browser during display of the page" in steps 72-76 of Figure 4. There is no viewer mentioned or suggested in steps 72-76 of Figure 4. Instead, a browser receives mouse coordinates in response to fielding a mouse-click event, and sends the mouse-click coordinates to the *server*.

Thus, of the features attributed by the Examiner to Guedalia, only the fact that Guedalia discloses a client-side browser requesting an HTML page from a server, and the server responding with the requested HTML page, are disclosed in Guedalia. Guedalia is unrelated to remaining features attributed by the Examiner to Guedalia.

The Examiner states in the Office Action that Beri discloses "an invocation of a viewer in the description of a web page sent to a client computer for display a requested image." Indeed, Beri does disclose the invocation of two different viewers – one for an initial HTML document that includes a <marquee> tag, and another within an HTML document displayed by the marquee object.

The Examiner states in the Office Action that Beri discloses "instantiating a viewer by a client computer for display an image included in the description of the web page." Apparently, the Examiner wishes to classify the marquee object as a viewer, although Beri is careful to classify only HTML document viewers as viewers. The marquee object invokes a second HTML viewer to actually display an image within the marquee window. It is not,

itself, a viewer, but instead controls the HTML viewer to display an image in a marquee-like fashion.

The Examiner states in the Office Action that Beri discloses "determining an action specified for performance in response to the input event to the action region and calling for performance of the determined action when the viewer determines that the input event was input to a position within the image corresponding to the active region." Beri discloses, in the first cited portion, column 6, lines 47-57, an attribute or parameter that can be passed, during invocation, from an HTML viewer to the instantiated marquee object (Beri, column 5, lines 37-61 and column 6, lines 47-57). This has nothing at all to do with determining an action in response to an input event. Beri discloses, in the second cited portion, column 6, line 66 to column 7, line 7, that a marquee object passes events to an HTML viewer, which runs as part of the browser. The browser receives the event, and then invokes a corresponding script, if one has been associated with the appropriate even handler. However, claim 1 explicitly states "when an input event is *detected by the browser* during display of the page, *passing the input event by the browser to the viewer.*" Thus, Beri discloses an event handling technique *opposite* from that claimed in claim 1. In the claimed invention, the browser detects the input event, and the input event is passed by the browser to the viewer. In Beri, the viewer (if the marquee object is considered to be a viewer) detects the event, and passes the event to the browser.

The Examiner states in the Office Action that Beri discloses "storing by the viewer representation of active regions within the image in image-relative coordinates along with indications of the actions to be performed in response to input events directed to the active regions." Beri discloses nothing at all related to image-relative coordinates. The Examiner explains citing column 5, line 40 to column 6, line 41 as follows: "the numbers X,Y pixels to move of the marquee object are defined suggests the image-relative coordinates be stored along with the action of scrolling performed in response to the input event of clicking on the marquee window." The explanation is largely undecipherable, and is incorrect. Pixel-based coordinates are device-relative coordinates, not image-relative coordinates. There is absolutely no suggestion in Beri of employing image-relative coordinates, and, instead, Beri contains an explicit reference to device-relative coordinates.

Thus, of the features attributed by the Examiner to Beri, Beri discloses only the feature of "an invocation of a viewer in the description of a web page sent to a client

computer for display a requested image." Beri is unrelated to remaining features attributed by the Examiner to Beri.

In short, Guedalia and Beri, together, fall far short of disclosing or suggesting the invention claimed in claim 1. Claim 1 is again provided, below, with those portions of claim 1 neither disclosed nor suggested in Guedalia and Beri emphasized:

> 1. A method for associating an active region with a corresponding position within an image included in a page displayed by a browser running on a client computer, the method comprising:
>
> sending a request by the browser to a server for a description of a page that includes a specification of the image **and an associated client-side image map, the client-side image map specifying a shape, size, and location of the active region within the image and specifying actions to be performed in response to input events directed to the active region**;
>
> receiving from the server in response to the request a description of the requested page that includes an invocation of a viewer for displaying the image, **the invocation including parameters that describe** the image and **the client-side image map;**
>
> instantiating the viewer and **passing to the viewer the parameters** included in the invocation;
>
> **storing by the viewer representations of active regions within the image in image-relative coordinates along with indications of the actions to be performed in response to input events directed to the active region**; and
>
> when **an input event is detected by the browser** during display of the page,
>
> **passing the input event by the browser to the viewer**, and
>
> when the viewer determines that the input event was input to a position within the image corresponding to the active region, determining an action specified for performance in response to the input event to the active region and calling for performance of the determined action. (emphasis added)

The remaining claims either depend from claim 1, include language equivalent to that of claim 1 that is directed to aspects of the present invention neither taught, mentioned, or suggested by Guedalia, Beri, or Guedalia and Beri in combination, or depend from claims that include language equivalent to that of claim 1 that is directed to aspects of the present invention neither taught, mentioned, or suggested by Guedalia, Beri, or Guedalia and Beri in combination. For example, independent claim 11 recites "the client-side image map specifying a shape, size, and location of the active region within the image and that specifies actions to be performed in response to input events directed to the active region." Neither Guedalia nor Beri disclose or suggest a client-side image map specifying a shape, size, and location of the active region within the image and that specifies actions to be performed in response to input events directed to the active region. In Guedalia, all actions are determined by the server. In Beri, all actions are determined by a browser responding to events passed to

it by the marquee object. As another example, claim 11 recites determining, by the server, of "the capabilities for viewing pages provided by the browser running on the client computer." Neither Guedalia nor Beri teach, mention, or suggest determining capabilities of a browser by a server, and sending a client-side image map only when the browser is capable of displaying dynamic images. As another example, independent claim 18 recites "data structures on the client computer that store image-relative indications of the particular portion of the image associated with the active region and actions." No image-relative indications are stored on the client side by either Guedalia or Beri, and no data structures are mentioned or suggested in either reference.

No Suggestion or Motivation to Combine References

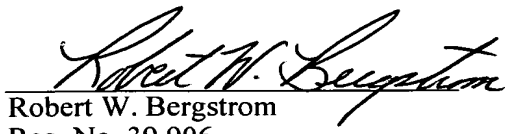
The Examiner states a rationale for combining Guedalia and Beri that Applicants' representative respectfully believes to be flawed. Guedalia explicitly states, multiple times, that it is Guedalia's intent to provide clientless image navigation (Guedalia, column 4, lines 3-5), so that a standard browser can be used on the client without additional software or modifications (Guedalia, column 4, lines 52-55). Guedalia is directed to display of images archived on a server, the images intentionally stored on the client (Guedalia, column 1, lines 45-47). There is no suggestion in Guedalia for moving navigational functionality, such as pan and zoom operations, from the server to the client, and, instead, Guedalia explicitly states the opposite. There is, therefore, no suggestion for adding client-side functionality from Beri to Guedalia's system. Moreover, Beri discloses a means for displaying a moving, animated image in a marquee window, while Guedalia is directed to navigating about a large, static image. These are completely dissimilar functionalities. Appellant's representative can see no workable combination of Guedalia and Beri, and Guedalia explicitly teaches away from the client-side techniques disclosed by Beri. Displaying a large, server-archived image that Guedalia seeks to navigate from a client computer in a scrolling marquee window would not provide any desirable functionality, and scrolling images within a marquee window do not contain active regions, in general, because a user would have to synchronize mouse movement with the scrolling image in order to access an active region. Moreover, a scrolling marquee window would be a decidedly inappropriate vehicle for display of portions of a large, static, archived image that one might want to selectively enlarge portions of for closer inspection.

CONCLUSION

The Examiner has cited a two references, Guedalia and Beri, in support of a rejection of claims 1-24 under 35 U.S.C. § 103(a). However, neither Guedalia nor Beri disclose, teach, mention, or suggest the claimed client-side image maps, representation of active regions within an image in image-relative coordinates by a client-side viewer, and many other claimed features of the currently claimed invention Guedalia and Beri are directed to quite different functionalities, and there is no obvious advantage in combining them, or even a workable combination. Displaying a large, server-archived image that Guedalia's disclosed method navigates from a client computer in Beri's scrolling marquee window would not provide any desirable functionality. For example, a user would have to synchronize mouse movement with the scrolling image in order to access a particular active region within the image. Moreover, a scrolling marquee window would be an inappropriate vehicle for display of portions of a large, archived image that one might want to selectively enlarge for closer inspection. There is no obvious or workable combination, or suggestion for combination.

Applicants respectfully submit that all statutory requirements are met and that the present application is allowable over all the references of record. Therefore, Applicants' respectfully requests that the present application be passed to issue.

Respectfully submitted,
Lawrence Bain et al.
OLYMPIC PATENT WORKS PLLC

By 
Robert W. Bergstrom
Reg. No. 39,906

Olympic Patent Works PLLC
P.O. Box 4277
Seattle, WA 98104
206.621.1933 telephone
206.621.5302 fax

APPENDIX I

1. A method for associating an active region with a corresponding position within an image included in a page displayed by a browser running on a client computer, the method comprising:

sending a request by the browser to a server for a description of a page that includes a specification of the image and an associated client-side image map, the client-side image map specifying a shape, size, and location of the active region within the image and specifying actions to be performed in response to input events directed to the active region;

receiving from the server in response to the request a description of the requested page that includes an invocation of a viewer for displaying the image, the invocation including parameters that describe the image and the client-side image map;

instantiating the viewer and passing to the viewer the parameters included in the invocation;

storing by the viewer representations of active regions within the image in image-relative coordinates along with indications of the actions to be performed in response to input events directed to the active region; and

when an input event is detected by the browser during display of the page,

passing the input event by the browser to the viewer, and

when the viewer determines that the input event was input to a position within the image corresponding to the active region, determining an action specified for performance in response to the input event to the active region and calling for performance of the determined action.

2. The method of claim 1 wherein the page displayed by the browser running on a client computer is a web page.

3. The method of claim 2 wherein the server runs on a server computer and a description of the web page is requested by the browser from the server and received by the browser from the server via the Internet.

4. The method of claim 2 wherein the server runs on the client computer and a description of the web page is requested by the browser from the server and received by the

browser from the server via an inter-process communications medium within the client computer.

5. The method of claim 2 wherein the description of the web page received from the server in response to the request by the browser is a hyper-text markup language document.

6. The method of claim 2 wherein the image is an OpenPix image and wherein an invocation to a browser extension image viewer is included in the description of the web page.

7. The method of claim 2 wherein input events directed to the active region may include mouse-click, mouse-into, and mouse-out-from events, and actions to be performed in response to input events include display of a web page, display of an image, or launching of a software routine.

8. The method of claim 2 where image-relative coordinates represent the position of points within the image, a point within the image represented by a pair of coordinates, a first coordinate of the pair having a fractional value representing the ratio of a horizontal line segment to a horizontal dimension of the image with a first endpoint coincident with a vertical edge of the image and a second endpoint coincident with the point, the horizontal line segment perpendicular to the vertical edge of the image, the second coordinate of the pair having a fractional value representing the ratio of a vertical line segment to a vertical dimension of the image with a first endpoint coincident with a horizontal edge of the image and a second endpoint coincident with the point, the vertical line segment perpendicular to the horizontal edge of the image, the horizontal and vertical edges of the image intersecting at an origin having coordinates (0, 0).

9. The method of claim 2 further including:

when a display altering input event is detected by the browser,

passing a display altering input command by the browser to the viewer, and

altering the display of the image by the viewer in accordance with the input command.

10. The method of claim 9 wherein display altering input events include a zoom input event and a pan input event.

11. A method for serving a description of a page from a server to a browser running on a client computer that requests the page, the description of the page provided to the browser by the server containing an invocation of a viewer, the invocation including parameters that specify an image included in the page and an active region within the image, the method comprising:

receiving a request from the browser by the server for a description of the page that includes a specification of the image and an associated client-side image map, the client-side image map specifying a shape, size, and location of the active region within the image and that specifies actions to be performed in response to input events directed to the active region;

retrieving a description of the page;

determining the capabilities for viewing pages provided by the browser running on the client computer; and

when the browser, running on the client computer, is capable of accepting display altering commands from a user while displaying a page,

parsing the description of the page to find the specification of the image and the client-side image map included in the page,

substituting, in the description of the page, an invocation of a viewer for the specification of the image and the client-side image map included in the page, including in the invocation parameters that specify the image and the client-side image map, to create a transformed page description, and

sending the transformed page description to the browser.

12. The method of claim 11 wherein the page requested from the server by the browser running on a client computer is a web page.

13. The method of claim 12 wherein the server runs on a server computer and a description of the web page is requested by the browser from the server and sent by the server to the browser via the Internet.

14. The method of claim 12 wherein the server runs on the client computer and a description of the web page is requested by the browser from the server and sent by the server to the browser via an inter-process communications medium within the client computer.

15. The method of claim 12 wherein the description of the web page retrieved by the server in response to the request by the browser is a hyper-text markup language document.

16. The method of claim 12 wherein the image is an OpenPix image and wherein an invocation to a browser extension image viewer is included in the description of the web page.

17. The method of claim 12 wherein input events directed to the active region may include mouse-click, mouse-into, and mouse-out-from events, and actions to be performed in response to input events include display of a web page, display of an image, or launching of a software routine.

18. A system for displaying a page that includes an image and an active region correlated with a particular portion of the image, the display of the page modifiable during the display of the page on a display device of a client computer such that the active region within the image remains correlated with the portion of the image, the system comprising:

a browser running on the client computer that displays the page;

a server that receives a request from the browser for a description of the page and that that provides a description of the page that contains an invocation of a viewer, the invocation including parameters that specify an image included in the page and an active region within the image; and

data structures on the client computer that store image-relative indications of the particular portion of the image associated with the active region and actions and actions to be performed in response to input events directed to the active region.

19. The system of claim 18 wherein the page displayed by the browser running on a client computer is a web page.

20. The system of claim 19 wherein the server runs on a server computer and a description of the web page is requested by the browser from the server and received by the browser from the server via the Internet.

21. The system of claim 19 wherein the server runs on the client computer and a description of the web page is requested by the browser from the server and received by the browser from the server via an inter-process communications medium within the client computer.

22. The system of claim 19 wherein the description of the web page received from the server in response to the request by the browser is a hyper-text markup language document.

23. The system of claim 19 wherein input events directed to the active region may include mouse-click, mouse-into, and mouse-out-from events, and actions to be performed in response to input events include display of a web page, display of an image, or launching of a software routine.

24. The system of claim 19 where image-relative coordinates represent the position of points within the image, a point within the image represented by a pair of coordinates, a first coordinate of the pair having a fractional value representing the ratio of a horizontal line segment to a horizontal dimension of the image with a first endpoint coincident with a vertical edge of the image and a second endpoint coincident with the point, the horizontal line segment perpendicular to the vertical edge of the image, the second coordinate of the pair having a fractional value representing the ratio of a vertical line segment to a vertical dimension of the image with a first endpoint coincident with a horizontal edge of the image and a second endpoint coincident with the point, the vertical line segment perpendicular to

the horizontal edge of the image, the horizontal and vertical edges of the image intersecting at an origin having coordinates (0, 0).